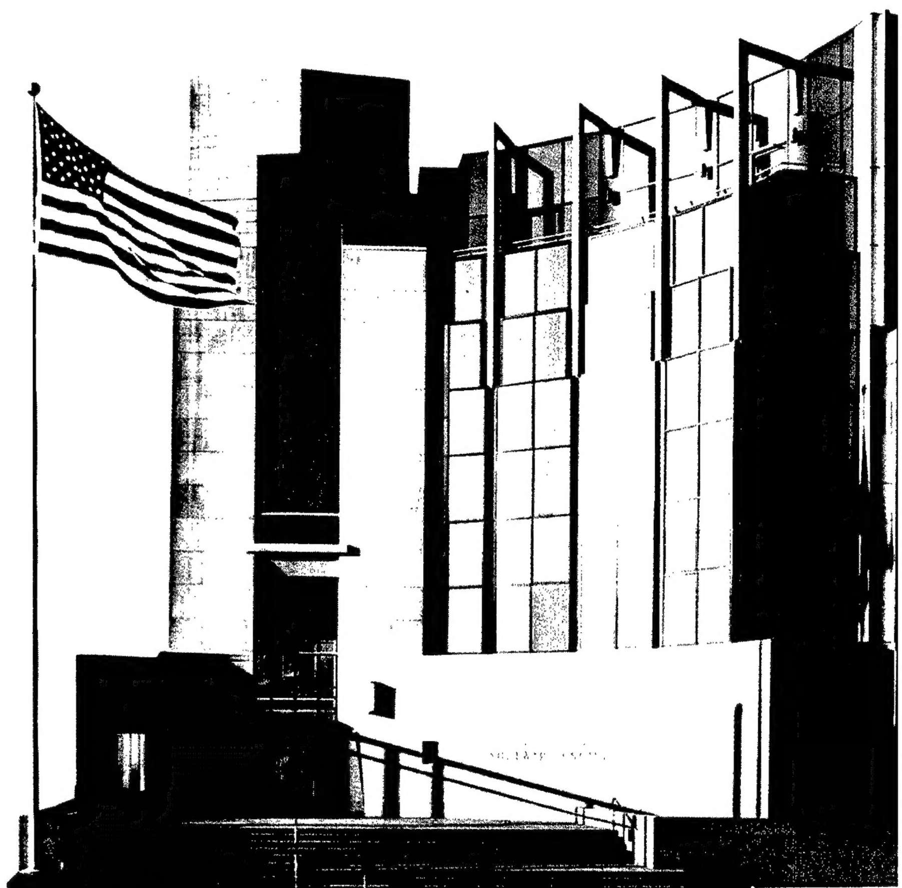
Carnegie Mellon
Software Engineering Institute

Current Perspectives on Interoperability

Lisa L. Brownsword
David J. Carney
David Fisher
Grace Lewis
Craig Meyers
Edwin J. Morris
Patrick R. H. Place
James Smith
Lutz Wrage

March 2004

TECHNICAL REPORT
CMU/SEI-2004-TR-009
ESC-TR-2004-009



Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Current Perspectives on Interoperability

CMU/SEI-2004-TR-009
ESC-TR-2004-009

Lisa L. Brownsword
David J. Carney
David Fisher
Grace Lewis
Craig Meyers
Edwin J. Morris
Patrick R. H. Place
James Smith
Lutz Wrage

March 2004

Integration of Software-Intensive Systems Initiative

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Christos Scondras
Chief of Programs, XPK

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Abstract

This report describes current research within the software engineering community on the topic of interoperability between software systems. That research includes analyses of the different types of interoperability problems and issues and efforts to define models of interoperability that will aid in creating solutions to those problems.

The report also describes work that is currently underway at the Software Engineering Institute (SEI) in this area. That work originated in an independent research effort and now has grown into a separate technical initiative in the area of interoperability. The SEI initiative is currently focused on analyzing several aspects of interoperability: how it is manifest in different kinds of activities (i.e., programmatic vs. constructive vs. operational activities), the essential characteristics of interoperability, and the key principles on which solutions will depend.

1 Introduction

1.1 Background

There is increasing demand for interoperability between individual software systems. This is true in almost every domain of computer software. In the public sector, interoperability between systems used by local, state, and federal health and law enforcement agencies is critical for homeland security. The military now relies on information sharing between air, sea, and ground-based forces to improve tracking of potential threats. Commercial organizations are trying to increase interoperability between their business systems to improve e-business efficiency, and at the same time are integrating dozens of microprocessors within products like automobiles. For example, the new Audi A8 has five main bus systems that connect the 60 computers controlling the drive train, the infotainment system, and comfort/convenience functions [Audiworld 02, Wilson 03]. Achieving interoperability among legacy systems as well as between legacy and new systems has become a driving factor for e-business, data warehousing, Web services, future military systems (e.g., Air Operations Center, Future Combat Systems), and other government systems (e.g., interoperability between local public health organizations and local police for homeland security).

To meet this increasing demand, organizations are attempting to migrate existing individual systems that employ disparate, poorly related, and sometimes conflicting systems to more cohesive systems that produce timely, enterprise-wise data that are then made available to the appropriate users. Meeting this goal has often proven to be considerably difficult.

1.2 Interoperability and Integration

There are many definitions for interoperability. Consider the following:

- the ability of two or more systems or components to exchange and use information (IEEE STD 610.12) [Standards 90]
- a) the ability of the systems, units, or forces to provide and receive services from other systems, units, or forces and to use the services so interchanged to enable them to operate effectively together
  b) the conditions achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users

c) the capacity to integrate technology between or among different technical platforms. This form of integration is achieved through information engineering, which translates process requirements into software programs (Joint Pub 1-02) [DoD 01].

- the ability to exchange data in a prescribed manner and the processing of such data to extract intelligible information that can be used to control/coordinate operations (FED-STD- 1037C) [NCS 96]

For our purposes, we need no more than a general working definition. In its most basic sense, interoperation requires some form of interchange between two or more entities. The nature of the entities may dictate the medium of the interchange but does not change the requirements of that interchange. While the distinction between interchange of information and interchange of services may be significant, services can only be interchanged if information can be interchanged, and interchange of information leads to interchange of services.

These observations lead us to a more abstract working definition of interoperability:

> The ability of a collection of communicating entities to (a) share specified information and (b) operate on that information according to an agreed operational semantics.

This definition is intended to be encompassing. The communicating entities can be people, computer systems, or a mixture of both. The shared information may be in the form of data or descriptions of services provided or capabilities required. The ability to operate on data according to agreed semantics is a fundamental requirement for interoperability between two systems that goes beyond the mere exchange of that data.

Interoperability is sometimes distinguished from integration, but at other times the two terms are used almost interchangeably. Dictionary definitions suggest that any significant difference between them lies in the degree of coupling between the entities. Thus, an integrated system is sometimes considered to be more tightly coupled than a system composed of interoperable components. Yet even this distinction suggests that perspective is a key factor in discussing interoperability. Thus, when looked on from a distance, a system is perceived to be integrated, but from the perspective of its constituent elements, they are interoperating with each other. The issue of perspective is recursive, because the interoperable entities themselves may be an integration of other constituents. Thus, the relationship of the observer to the constituent makes a difference as to whether the appropriate term is integration or interoperability. We shall not make any further distinction between these terms in the remainder of this report.

1.3 Scope of this Report

The report is written in the context of a growing community of interest in interoperability. Many programs and initiatives have begun to deal with its myriad of problems; we shall discuss several of these initiatives in Chapter 3. Some of those initiatives are strongly technical

in nature, while others are more focused on long-term strategic issues. In this report, we focus primarily on software issues. However, program managers faced with the creation of systems that must interoperate with others may find benefit from certain elements of this report (e.g., the characteristics of interoperability). Other readers engaged in various activities related to interoperable systems may also find some benefit from the discussions of the problems and solutions, if only to learn lessons from others.

This report describes the intellectual foundation of the work of the Carnegie Mellon® Software Engineering Institute (SEI), specifically of the Integration of Software-Intensive Systems (ISIS) initiative. In Chapter 2, we discuss the problem space of interoperability, particularly with regard to the U.S. Department of Defense (DoD) domain. In Chapter 3 we discuss the potential solution space by examining other initiatives and approaches to interoperability that are either currently being employed or are being recommended for near-term implementation. We also discuss various models of interoperability current in the software community. In Chapter 4 we describe our own perspective on the interoperability problem space, and propose several principles of interoperability. We also suggest a model of categories of interoperable issues, and discuss some of the characteristics of any interchange that are necessary to achieve interoperability. These characteristics may be used to measure solutions (both procedural and technological) as well as to provide a more detailed understanding of interoperability. In Chapter 5 we summarize the main points of this report and describe our plans for work in this area.

2 A View of the Problem Space

It is difficult to achieve the degrees of interoperability currently being sought—this has been demonstrated by ample evidence.[1] Whether the goal is to increase interoperability between existing, standalone systems or to build complex new systems designed to interoperate, the challenge is not a simple one.

In this chapter, we will consider some of the factors that make interoperability such a complex issue. We do so first through summarizing recent SEI research into interoperability problems encountered in DoD programs. We then examine two simple scenarios, both of which are based on actual examples that were described as part of the research study; these scenarios illuminate some of the engineering and organizational problems in practice. Note that we will further consider our perspectives on the interoperability problem space in Chapter 4.

2.1 SEI Study of Interoperability Problems

A recent SEI study focused on identifying current problems in interoperability within DoD systems [Levine 02]. All of the problems were collected from interviews and workshops with personnel from the DoD.

The study indicated that new systems designed and constructed to interoperate with existing and other new systems, and adhering to common standards, still fail to interoperate as expected. Reasons for the failures vary, but incomplete requirements, unexpected interactions, and unshared assumptions were common. Among the more specific interoperability problems that were identified are the following:

- Planned interoperability between new systems is often scaled back in order to maintain compatibility with older systems that cannot be upgraded without major rework.

---

1 See, for example, references to Bluetooth interoperability:
http://www.mobile.commerce.net/story.php?story_id=2117&s=4
Verizon Wireless interoperability:
http://www.mks.com/products/pdfs/MKS_casestudy_VerizonWireless.pdf
and Military Coalition interoperability:
http://www.mitre.org/work/tech_papers/tech_papers_03/skidmore_coalition_interop/skidmore_coalition_interop.pdf

- Strict specification of standards like Link-16[2] proves insufficient for achieving desired levels of interoperability, because organizations constructing "compliant" systems interpret specifications in different ways, thus creating different variants of the links.
- Policies promote a single point of view at the expense of other points of view. For example, policies that enhance the levels of interoperability that can be achieved in one domain are generalized to additional domains, where they unduly constrain organizations trying to produce interoperable systems.
- Funding and control structures within the DoD are not providing the incentives necessary to achieve interoperability. For example, even though there is increased emphasis on the joint nature of many systems, funding for most programs still flows through service sponsors.
- Tests constructed to verify interoperability frequently fail to identify interoperability shortfalls. In other cases, systems are approved for release in spite of failing interoperability tests.
- Even when interoperability is achieved by systems of systems, it is difficult to maintain as new versions of constituent systems are released. New system versions frequently break interoperability.

To see examples of interoperability problems in practice, we now examine two scenarios, both drawn from actual programs, that encountered severe difficulties in achieving interoperability between multiple, independent systems.

2.2 Scenario 1: DoD Tracking Systems

One DoD organization developed a system to identify and track potentially hostile aircraft and to control the movement of friendly aircraft. Another DoD organization developed a system to track and destroy threatening incoming objects, such as missiles. After these two systems were fielded, it became apparent that the overall capability of both systems could be enhanced by sharing information regarding objects being tracked by the two systems. However, even though the systems performed similar tracking missions, the desired degree of interoperability could not be achieved because the characteristics of the data maintained (e.g., accuracy, frequency of refresh) differed between the systems. As a result, system operators for the system that maintained higher fidelity data were required to learn how to use potentially "degraded" data received from the other system.

This scenario highlights some critical problems:

- Future needs for interoperability are often unknown at the time that individual systems are specified. In this case, the value of interoperability between the two tracking systems

2 See http://jitc.fhu.disa.mil/gccsiop/interfaces/link16.htm.

was only discovered after both were fielded and when the services began to emphasize joint operations. In general, no matter how carefully requirements for individual systems are identified, situations for potential benefits from interoperation will arise that were not envisioned. We use the term *opportunistic interoperability* for this kind of scenario.

- Even when systems share a common function (in this case, tracking of air objects), differences in data models and assumptions about how that data will be used can limit interoperability.
- Solutions to interoperability problems are not exclusively technical. In this case, the solution involved both technical changes to the systems and changes to the operational procedures used by end users.

2.3 Scenario 2: Banking Consolidation

After deregulation of the banking industry, a commercial bank built new capabilities in financial management, insurance, and other areas by acquiring a variety of other firms. Divisions of the company (previously independent firms) were treated as independent cost centers to encourage retention of staff and motivate high achievement. Following the consolidation cycle, the now larger institution recognized that it had acquired dozens of information systems (e.g., budget, personnel) that provided inconsistent, but essentially duplicate, services. These systems ran on diverse platforms, used different databases, and had unique user interface and communication features. To become more competitive, the institution would have to consolidate some systems and develop mechanisms for interoperation between others. Unfortunately, initiatives to identify a common architecture for computing infrastructure and information management were troubled due to the differing demands of the managers of the various divisions. Division managers, who maintained significant independence after the merger, were incentivized to minimize cost and disruption to their existing computing infrastructure and systems.

This scenario highlights several additional problems:

- Interoperability problems can involve many systems and may not be effectively solved by developing point-to-point solutions between pairs of systems. This is particularly the case for newly conceived systems of systems such as the Army's Future Combat System (FCS) and the systems that will be built to support net-centric warfare.
- Organizations are driven by many often competing incentives that may be at odds with achieving interoperability goals. In this case, potential cost and disruption served as a strong disincentive to making compromises necessary for interoperability. In general, independent organizations, such as distinct corporations or the various branches of the military (to be more precise, the people within those organizations), are influenced by many incentives that discourage interoperability.

The air tracking and banking scenarios, as well as the results of our research effort, show a number of clear interoperability problems and hint at several underlying causes. In Chapter 3 we summarize some current efforts to resolve these problems. Note that in Chapter 4 we will revisit the discussion of the problem space in the context of the SEI's perspectives on interoperability.

3 Current Approaches to Achieving Interoperability

This section will briefly examine some of the numerous programs, initiatives, and organizations exploring various solutions to "the interoperability problem." We will also discuss the implications of various interoperability models and technological approaches to the goal of achieving interoperability.

3.1 Current Interoperability Initiatives

Most current initiatives tend to focus either on developing practices that improve interoperability or on developing models of interoperability that will provide a basis for such improvement. The former tend to fall into the following categories:

- requirements improvement
- standards
- synchronization of system schedules
- operational issues

We will briefly consider examples of each of these. We will then consider at greater length the initiatives that are aimed at models of interoperability.

3.1.1 Requirements Improvement

Initial efforts at enhancing interoperability focused on improving the requirements process to describe the desired interoperability. Requirements improvement usually takes the form of examining requirements for interoperability and harmonizing requirements across multiple, cooperating systems. The Goldwater-Nichols Act of 1986 mandated jointness, among other things, and established interoperability as a requirement for military command and control systems [Goldwater 86]. Since then, it has been common for Operational Requirements Documents (ORDs) to include lists of systems or capabilities with which the subject system must interoperate and standards with which it must comply. At the ORD level, interoperability requirements are developed and approved through the involvement of a number of organizations, including the Joint Requirements Oversight Council (JROC), the Joint Forces Command (JFCOM) Joint Interoperability and Integration (JI&I) Directorate, the Joint Re-

quirements and Integration Directorate of the Joint Chiefs of Staff (J8), and the Combatant Command Interoperability Program Office.

3.1.2 Standards-Based Approaches

A common component of virtually all strategies for improving interoperability is the requirement to conform to some set of standards. Standards conformance has long been a staple of defense acquisition, although their use has undergone significant changes in the past 10 years. Rather than specifying how to build systems, requirements compliance is now used as a means to leverage commercial technologies and manufacturing processes while meeting military-unique environmental and operational requirements. Examples of this include requirements to conform to the (mostly commercial) standards specified in the Joint Technical Architecture (JTA), and certification for operation with the Common Operating Environment (COE). More recently, and mirroring a similar push in the commercial marketplace, there has been strong advocacy for the use of the eXtensible Markup Language (XML) as a *lingua franca* for achieving interoperability; this movement is gaining momentum as the DoD migrates from the COE architecture to one based on the Net-centric Core Enterprise Services (NCES). Related efforts, like Defense Information Systems Agency's (DISA) DoD XML Registry and Clearinghouse and the Universal Data Element Framework (UDEF), are attempting to define common XML components and data/metadata interchange definitions for use by all programs.

Test facilities (e.g., engineering laboratories and simulators) have taken on increased importance in assessing standards-based interoperability. While such facilities have long been used in achieving a degree of interoperability (e.g., the Navy's Integrated Combat System Test Facility, ICSTF), there has recently been a significant growth in demands on such environments to support large-scale, high-fidelity integration and interoperability engineering. Virtual facilities, like the Navy's Distributed Engineering Plant, make it possible to identify present and reasonably foreseeable interoperability problems, facilitating resolutions well in advance of actual system integration and deployment.

3.1.3 System Synchronization

An increasingly popular approach to achieving interoperability is to synchronize system development and deployment through some form of "blocking" strategy, where fixed deployment epochs provide interoperability milestones that must be met by all systems prior to deployment. Examples of this approach include the Navy D-30 Certification process long used for battlegroup deployments and the Army Software Blocking strategy recently implemented in the Future Combat System (FCS).

More recently, a hybrid of these two approaches has emerged and is illustrated by the tactic used in some large joint programs, wherein individual service requirements are subordinated

to the joint warfighting requirements. Thereafter, service development and acquisition schedules are aligned across system and service boundaries to permit the use of greater commonality between services/systems. This leads to DoD-wide cost reductions—at potentially higher costs, or delays in fielding—for an individual system or service. Examples of this approach include the Joint Strike Fighter (JSF) program and, more recently, the re-aligned Joint Tactical Radio System (JTRS) program.

3.1.4 Operational Approaches

Another widely used method for improving interoperability is to explore the operational aspects of a complex system, such as a naval battle group or armored division, through the use of exercises, tests, and simulations. JFCOM was recently given the responsibility (as the DoD's "transformation laboratory") to develop concepts, test those concepts through experimentation, and train the joint leadership. Some other organizations active in this area include the DISA Interoperability Directorate, the Joint Interoperability Test Command (JITC), and the Joint C4ISR Battle Center. Every two years, the Joint Chiefs of Staff sponsor the Joint Warrior Interoperability Demonstration (JWID), with a goal of identifying promising technologies and operational concepts for further refinement.

3.2 Current Models of Interoperability

Model-based approaches have proven valuable in almost every engineering domain. Within the domain of software engineering, architecture models are currently being extensively exploited, and efforts in software cost modeling, particularly with regard to commercial software components, are underway in many organizations.

There are also several ongoing attempts to develop useful models of software interoperability. The following sections examine some of the current modeling efforts, with an emphasis on those with particular significance for the DoD. In Chapter 4, we shall propose some concepts of an interoperability model that expand on these efforts.

3.2.1 NATO C3 Technical Architecture (NC3TA) Reference Model for Interoperability (NMI)

The North Atlantic Treaty Organization (NATO) defined an interoperability model (NMI) as

> *The minimal set of rules governing the specification, interaction, and interdependence of the parts or elements of NC3 Systems whose purpose is to ensure interoperability by conforming to the technical requirements of all NC3TA Volumes. The NC3TA identifies the services, building blocks, interfaces, standards, profiles and related products and provides the technical guidelines for implementation of NC3 Systems* [NATO 00].

In the NMI model, the notion of interoperability is essentially concerned with data. The model uses five levels, each of which provides an increasing degree of data sharing between two systems:

- **No Data Exchange**: no physical connection
- **Unstructured Data Exchange**: exchange of human-interpretable, unstructured data (free text)
- **Structured Data Exchange**: exchange of human-interpretable structured data intended for manual and/or automated handling, but requires manual compilation, receipt, and/or message dispatch
- **Seamless Sharing of Data**: automated data sharing within systems based on a common exchange model
- **Seamless Sharing of Information**: universal interpretation of information through cooperative data processing

3.2.2 Levels of Information Systems Interoperability (LISI) Model

LISI is a product of the C4ISR Working Group sponsored by the DoD. It was begun in 1993, and the most recent version of the model was released in 1998. The LISI model is focused on the notion of "maturity" of interoperability; it depicts five levels of interoperability, each of greater maturity:

- **Isolated Systems**: No physical connection exists.
- **Connected Systems**: Homogeneous product exchange is possible.
- **Distributed Systems**: Heterogeneous product exchange is possible.
- **Integrated Systems**: Shared applications and shared data exist.
- **Universal Systems**: Enterprise-wide shared systems exist.

These levels have some similarity with those of the NMI model. In addition, however, LISI distinguishes four "attributes" of interoperability: Procedure, Applications, Infrastructure, and Data (PAID). These attributes "encompass the full range of interoperability considerations. They assist in defining the sets of characteristics for the exchange of services at each level of sophistication" [C4ISR 98]. Taken together, the levels and attributes enable the LISI model to define a profile of interoperability illustrated in Figure 1.

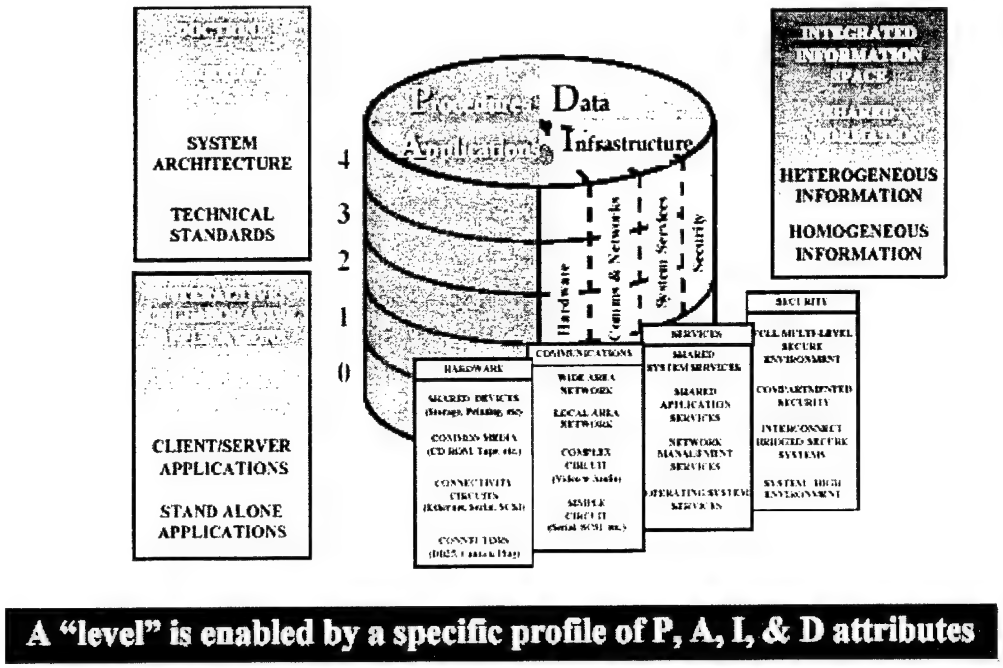


Figure 1: The LISI "PAID" Paradigm

The PAID attributes extend LISI away from a purely data-centric view of interoperability. However, even with these attributes as part of the model, there is nonetheless a basic conception of interoperability as rooted in technological phenomena, and interoperability at the data level is certainly a central aspect of LISI.

3.2.3 The "Levels of Conceptual Interoperability" Model (LCIM)

A third current model, developed by Tolk and Muguira, emerged from the needs of the modeling and simulation community. LCIM takes cognizance of the models discussed in the previous two sections. The rationale for the model is that

> *...although the unambiguous interpretation of the meaning of the data [e.g., as in NMI and LISI] to be interchanged between two systems is necessary, it is not sufficient....Establishment of metadata standards allows a much more open use of data within the systems, as not the data itself has to be standardized, but the interpretation of the data in the given context* [Tolk 03].

Like LISI and NMI, the LCIM model defines five levels:

- 0 – System-Specific Data: No interoperability exists between two systems; data are used in each system in a proprietary way with no sharing
- 1 – Documented Data: Data are documented using a common protocol (e.g., the HLA Object Model Template) and are accessible via interfaces.
- 2 – Aligned Static Data: Data are documented using a common reference model or metadata standards; meaning of data is unambiguously described.
- 3 – Aligned Dynamic Data: Use of data within the federate/component is well defined using standard software engineering methods such as UML. This shows the use of data within the otherwise unknown "black box behind the interface."
- 4 – Harmonized Data Semantic: Connections between data that are not related concerning the execution code are made obvious by documenting the conceptual model underlying the component.

While LCIM is based on different levels of data interoperability, the intention of the model is somewhat different from that of LISI and NMI:

> *While in the world of state-of-the-art C4ISR [the implicit context of NMI and LISI] many problems are solved...modeling and simulation systems deal...with the agile component of the battlefield....Meaningful interoperability of simulation systems requires composable models on the conceptual level* [Tolk 03].

Thus, according to Tolk and Muguira, to deal with the higher levels of their model, the conceptual model must examine semantic relations; achieving the highest level of interoperability between two systems can only be done by full alignment of conceptual models that underlie each of the two systems.

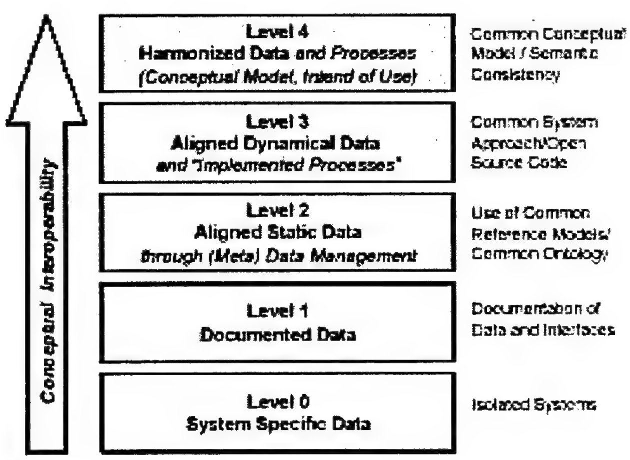


Figure 2: Levels of LCIM

3.2.4 The "Organizational Interoperability Maturity" Model

An effort by the Australian Ministry of Defense has extended the LISI model toward the notion of organization interoperability. In effect, the ministry conceives a different dimension for interoperability from that of data, positing a parallel structure to LISI but with the emphasis on human interaction. There are five levels and four kinds of attributes, as in LISI. The levels are Independent, Ad hoc, Collaborative, Combined, and Unified, and the four organizational attributes are Preparedness, Understanding, Command style, and Ethos. All of these center on the interrelationships between people and organizations [Warner 02]:

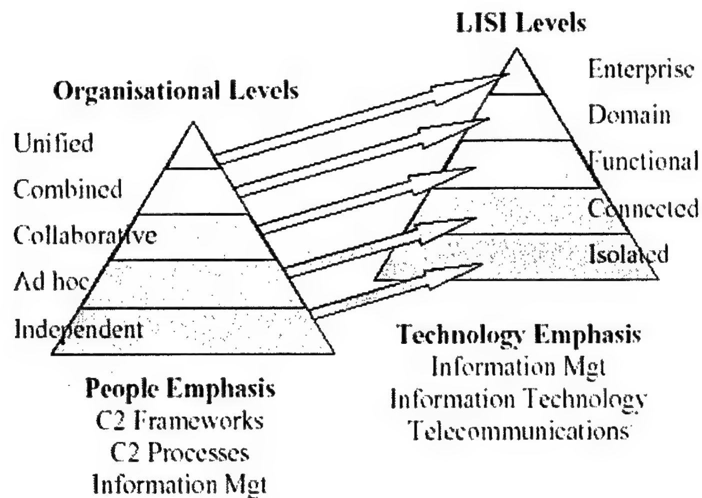


Figure 3: Parallels between Organizational Model and LISI

This model is seen as a companion to the LISI model; it also has a focus on "the formation and operation of joint, allied, and coalition deployable task forces."

3.3 Analysis of Existing Approaches

While the preceding are not inclusive of all current initiatives in interoperability, they represent a reasonable selection. And in spite of a considerable body of work that has been done over the past several years, the problems of achieving improvements in interoperability — whether through improved practices directly, or by creating a cohesive and widely accepted model—seem as difficult as ever.

As described in Chapter 2, the SEI recently performed a study to investigate ways in which interoperability problems were being solved in the DoD and in other government programs. In addition to identifying the problems that currently exist, another goal of the study was to identify solutions or partial solutions that have been used, as well as to identify ways in which the SEI can contribute to solving problems dealing with interoperability.

We have already listed the most significant problems (see pages 4-5). Together with DoD personnel, the research team also analyzed these problems and determined that the following general themes are the keys to solving these problems:

- Complexity is caused by many problems and many players.
- Interoperability is more than a technical problem.
- Funding and control are not aligned.
- Leadership, direction, and policy are not effective.
- Legacy systems are a persistent problem.

Above all, the research effort determined that, based on the aggregate experience of all who contributed to the study, the bulk of issues lie in the domain of management rather than in the sphere of technology. While not minimizing the technical problems inherent in integrating two or more complex systems, it is the management and programmatic aspects of interoperabilty that have led to many costly failures.

This indicates that interoperability is a multi-dimensional phenomenon. Its scope certainly includes technical elements (as per the NMI and LISI models) and management elements (as per the Organizational Model), but also includes broader programmatic elements as well, such as harmonization of independent schedules for joint deployment and update. To understand interoperability in its widest sense, we must conceive and model all of its aspects and account for all of those elements—which may in fact be in competition—that can either bring about or prevent interoperability in any given instance.

We must also find ways to quantify interoperability. For instance, what does it mean if we say that two systems are "partially interoperable"? In the area of data sharing and connectivity, LISI and its companions make an excellent beginning by enumerating levels of data integration and defining standards that support them. But, as we have pointed out, interoperability exists at many levels, not merely data, and simply sharing data will not automatically produce interoperable systems.

In the following chapter, we propose some concepts that we believe will prove useful in finding solutions to "the interoperability problem." We believe that considerable future work is needed and expect that these concepts will prove foundational for that work.

4 Conceptual Foundation

In this chapter, we revisit some of the key problems of interoperability and put forth our perspective on those problems and their solutions. We first enumerate several principles that we believe must be the basis of any useful solutions. We then describe the System-of-Systems Interoperability (SOSI) model, which will inform our choices of specific interoperability problems to study and solutions to propose. Finally, we suggest an initial set of characteristics or attributes of interoperability that will play a major role in developing more detailed understanding of the range of processes, standards, technologies and tools needed to achieve the full degree of interoperability that is envisioned for the coming decades.

4.1 Guiding Principles

As Fred Brooks pointed out more than 15 years ago, the factors that make building software inherently difficult are complexity, conformity, changeability, and invisibility [Brooks 87]. With apologies to Brooks, we assert that achieving and maintaining interoperability between systems is also inherently difficult, due to

- complexity of the individual systems and of the potential interactions between systems
- lack of conformity between human institutions involved in the software process and resulting lack of consistency in the systems they produce
- changeability of the expectations placed on systems (particularly software) and the resulting volatility in the interactions
- invisibility of all of the details within and between interoperating systems

In spite of a considerable effort, technical innovations aimed at improving software engineering have not successfully attacked the problems represented by these essential characteristics. In fact, today's interoperating systems are likely more complex (due to the massive increase in the number of potential system-of-systems states) than those examined by Brooks. They exhibit less conformity (due to the increased diversity of the institutions involved in construction of the constituent parts), are more volatile (due to the need to accommodate widely diverse users) and have even poorer visibility (due to size, number of participating organizations, etc.) We therefore posit a set of six principles that will inform our efforts in the selection of problems to address and, more critically, in the analysis of potential solutions. The principles are

1. No clear distinction exists between *systems* and *systems of systems.*

2. Most interoperability problems are independent of domain.
3. Solutions cannot depend on complete information.
4. No one-time solution is possible.
5. New technologies constantly move systems toward legacy status.
6. Networks of systems demonstrate emergent properties.

Because we believe that any approach that ignores the consequences of these principles will fail, we discuss each of these principles in detail.

4.1.1 No Clear Distinction Between Systems and Systems of Systems

The distinction between a system and a *system of systems* is often unclear and seldom useful. By this we mean that many, perhaps a majority, of "systems" are actually systems of systems in their own right. The critical factor is less where a boundary might lie and more where control lies: most systems are now created with some components over which the integrator has less than complete control. Further, most systems must cooperate with other systems over which the integrator often has no control.

A good example is the U.S. Air Force F-22 Raptor "integrated" avionics system. This system consists of navigation, weapons stores management, mission and vehicle management, inertial reference, self-defense, instrumentation, and other components. Clearly, by some definition, each of these pieces represents a system in its own right, whether from the perspective of function, its creation, or the hardware platform it executes on. Equally clearly, each of these systems will be composed of other pieces, some commercially purchased, some obtained from other programs, and some from a custom implementation.

It is often stated that "What someone considers to be a system of systems, somebody else considers a system." Thus, for any given entity, one's perspective could see it as a component (of a larger system), as a system in itself, or as a system of systems. The Raptor's avionics system is certainly "a system." But from a perspective of joint operations, it is an F-22 Raptor that is "a single system," of which its avionics system is simply a component. And, more importantly, *there usually is no top level*, because inevitably there will be some demand to include any system of systems in a more encompassing system of systems.

4.1.2 Interoperability Problems Independent of Domain

In Chapter 2 we illustrated the myriad of problems that arise in building interoperable systems. The two scenarios we highlighted (e.g., creating interoperability between multiple business systems and sharing of air tracking information) indicate another important fact—

that interoperability problems exist for both business and combat systems. In fact, most complex systems in almost every domain are now expected to interact with other complex systems. Regardless of domain, interoperability problems persist, and the costs of failures are huge. As an example, within the U.S. auto supply chain, one estimate put the cost of imperfect interoperability at one billion U.S. dollars per year, with the largest component of that cost due to mitigating problems by repairing or reentering data manually [Brunnermeier 99].

Our expectations are for even greater degrees of interoperability in the future, a goal that may prove difficult to achieve. The current generation of interoperable systems at least tend to be knowledgeable participants in the interaction—that is, the systems are being designed (or modified) specifically to interact with a particular system (or limited set of systems) in a controlled manner, and to achieve predetermined goals. What is new about the future generations of interoperating systems is an emphasis on dynamically reconfigurable systems. These systems—or more accurately the services they provide—are expected to interoperate in potentially unplanned ways to meet unforeseen goals or threats.

Within the defense community, the most prominent example of this new type of interoperation is Net-Centric Warfare and the systems that will support it. A similar concept exists in the business community, where a major goal involves Web-enabled dynamic discovery of third-party business services that can be stitched together with existing core business services to create new capabilities.

We do not suggest that the solutions eventually found for the interoperability problems should be identical across domains. But we believe that the various communities should be aware of each other and look for commonality of high-level purpose and solution strategy—if not of solution detail—within other communities. In Chapter 6 we will consider some of the ways that such awareness might be achieved.

4.1.3 Solutions Cannot Rely on Complete Information

Classic software engineering practice assumes a priori understanding of the system being built, including complete and precise comprehension of

- assumptions or preconditions expected of the system that are required for successful use, including standards, system and environmental conditions, and data and interactions expected of other hardware, software, and users
- functionality, services, data, and interactions to be obtained from and provided to outside agents
- non-functional properties or quality of service required by the system and expected of the system from interacting components

For interoperable systems, the same information is required by all participants: the individual components (i.e., the individual systems), the links between them, and the composite system of systems. It would therefore seem that for an organization building a component (system), complete knowledge of all expectations is necessary to complete it. Unfortunately, we seldom (if ever) have such complete and precise specification even *when* a single system is *only* expected to operate in isolation.

The reality is that multiple organizations responsible for integrating multiple systems into interoperating systems of systems have multiple—and rarely parallel—sets of expectations about the constituent parts, as well as different expectations about the entire system of systems. The decisions that they make about the overall system of systems, e.g., assumptions, preconditions, functionality, and quality of service, are just as likely to be as incomplete and imprecise as those of organizations responsible for a single system.

Given that having complete and precise information about a system of systems (and its constituent parts) is not possible, two approaches to managing the potential chaos are evident:

- Reduce imprecision by enforcing common requirements, standards, and managerial control.
- Accept imprecision as a given and apply engineering techniques that are intended to increase precision over time, such as prototyping and spiral models of development.

The first approach alone may well increase interoperability to a significant degree, but it is also highly static and does not address the inherent imprecision in the software engineering process or the legitimate variation in individual systems. The second approach is limited in a different way, since without agreeing on some level of commonality, we are left with an "every system for itself" world that will not approach the levels of interoperability we require.

4.1.4 No One-time Solution Is Possible

We live in a dynamic and competitive world in which the needed capabilities of systems must constantly change to provide additional benefits, to counter capabilities of adversaries, to exploit new technologies, or in reaction to increased understanding or evolving desires or preferences of users. Simply put, systems must evolve to remain useful.

This evolution affects both individual systems and systems of systems. Individual systems must be modified to address unique and changing demands of their specific context and users. The expectations that systems of systems place on constituent systems will likewise change with new demands. However, the changing demands placed on a system by its immediate owners and those placed by aggregate systems of systems in which it participates are often not the same, and in some cases are incompatible.

The result is that maintaining interoperability is an ongoing problem. This was verified by our interviews with experts who had worked with interoperability. In some cases, desired system upgrades did not happen because of the impending effect on related systems. In other cases, expensive (often emergency) fixes and upgrades were forced on systems by changes to other systems.

In order to maintain interoperability, new approaches are needed to

- vet proposed requirements changes at the system and system-of-systems level
- analyze the effect of proposed requirements and structural changes to systems and systems of systems
- structure systems and systems of systems to avoid (or at least delay) the effect of changes
- verify interoperability expectations to avoid surprises when systems are deployed

New approaches to structuring systems that anticipate changes, that vet requirements and structural changes and analyze their impact, and that verify that systems of systems perform as anticipated will go a long way toward maintaining the interoperability of related systems.

4.1.5 New Technologies Move Systems Toward Legacy Status

One expert interviewed by the SEI fantasized about raising the technology floor of all systems in order to increase interoperability between systems. However much we would like to start over, it is not possible due to the cost, schedule, and disruption involved. This is true even though levels of interoperability that can be obtained will remain limited by existing systems.

However, it is useful to follow the fantasy further. Assuming for a minute that we could start over, all we would buy would be a momentary illusion of complete consistency between systems. During the time it took to recreate all of the necessary capabilities, both user expectations and computing technology would change—new capabilities would be required and new technologies developed. As a result, decisions would have to be made and remade regarding what features to implement and what new technologies to deploy. Each shift away from the baseline during any multi-year effort to raise the technology floor would create an entire new set of legacy problems. As expressed by one expert interviewed by the SEI, "today's cutting edge solution is tomorrow's legacy system."

4.1.6 Networks of Interoperability Demonstrate Emergent Properties

Emergent properties are those properties of a whole that are different from, and not predictable from, the cumulative properties of the entities that make up the whole. The core concept can be traced as far back as Plato, who proposed that the mind was a distinct entity and *not* an emergent property from the human body. More recent philosophers (and biologists, ecologists, computer scientists, etc.) have accepted both the concept and the validity of the concept when applied to their domains.

The concept of emergent properties becomes increasingly important as the number and type of "actors" in a system of systems increase. Thus, large-scale networks such as the Internet (and in the future, networks that support net-centric warfare) are likely to experience emerging properties. Such networks are composed of large numbers of widely varied components (hosts, routers, links, users, etc.) that interact in complex ways with each other.

Of necessity, each participant in such real-world systems (both the actor in the network and the engineer who constructed it) acts primarily in his or her own best interest. As a result, perceptions of system-wide requirements are interpreted and implemented differently by various participants, and local needs often conflict with overall system goals. Although collective behavior is governed by control structures (e.g., in the case of the networks, network protocols), central control can never be fully effective in managing complex, large-scale, distributed, or networked systems.

The net effect is that the global properties, capabilities, and services of the system as a whole emerge from the cumulative effects of the actions and interactions of the individual participants propagated throughout the system. The resulting collective behavior of the complex network shows emergent properties that arise out of the interactions between the participants.

The effect of emergent properties can be profound. In the best cases, the properties can provide unanticipated benefits to users. In the worst cases, emergent properties can detract from overall capability. In all cases, emergent properties make predictions about behavior such as reliability, performance, and security suspect.[3] This is potentially the greatest risk to wide-scale networked systems of systems. ISIS recognizes that any long-term solution must involve better understanding and managing of emergent properties.

3 There is a continuing debate as to whether a property is truly emergent or whether, as we learn more about the components and their interactions, we will begin to find techniques for prediction of properties that appear to be emergent. We are neutral with regards to this debate, but believe that we must begin to understand the nature of emergent properties in networked systems of systems and find ways to manage problematic behaviors that occur—whether we can predict them or not.

4.2 The SOSI Model of Interoperability

As noted in Chapter 3, the SEI's analysis of interoperability problems indicated that interoperability has many aspects, and achieving interoperability will require a broad range of improved practices. To understand interoperability fully, it is useful to model all of its aspects.

As part of the study, we developed a simple model, the System-of-Systems Interoperability (SOSI) model that depicts the broad range of activities within a single program that will impact any efforts to achieve interoperability. These activities are indicated in Figure 4. (Note that in this view of a program, the end user is considered to be part of the operational system.)

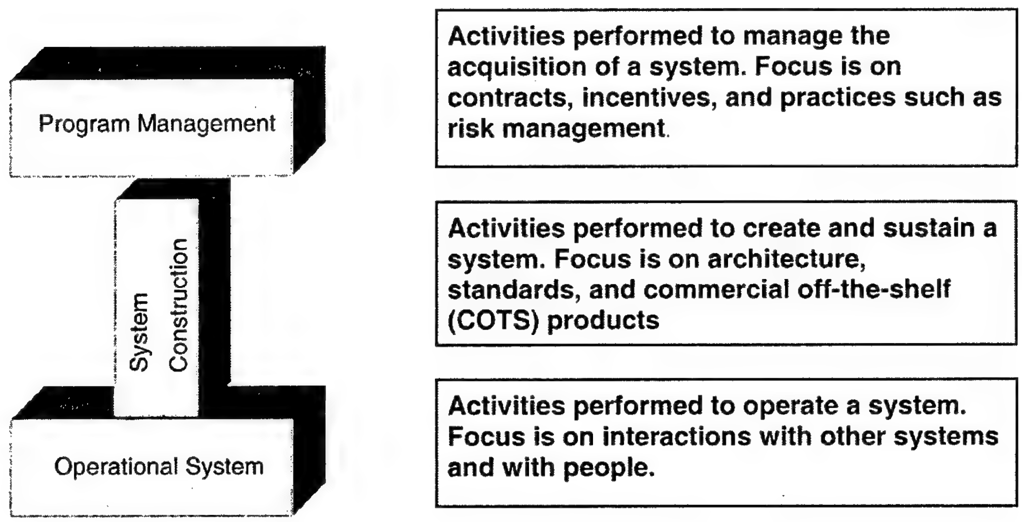


Figure 4: System Activities Model

Figure 4 indicates how this model is useful for portraying different kinds of interoperability, since interoperability between multiple systems requires interoperation both between programs and between the systems they control. We believe that this interoperability must occur in different ways at the level of the program management office (PMO), system construction, and operation. Each dimension represents a type of interoperability (programmatic, constructive, and operational). Each of these is discussed in greater detail on the next pages.

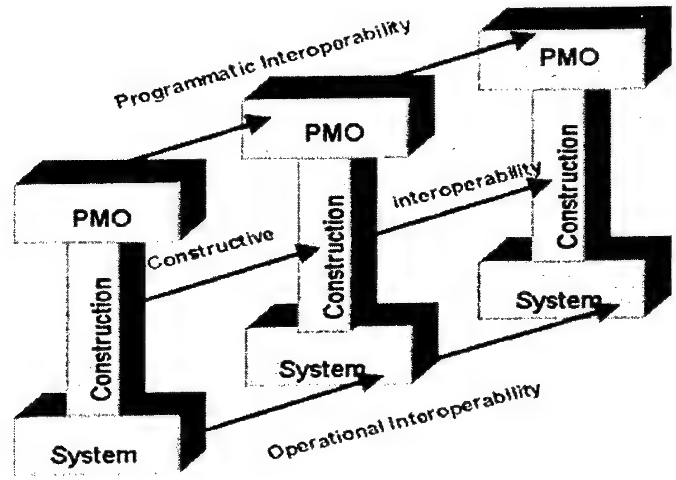


Figure 5: Different Dimensions of Interoperability

4.2.1 Programmatic Interoperability

Programmatic interoperability encompasses the activities related to the management of one program in the context of other programs. Typically, programs are managed in isolation, with little need to consider the management functions of other systems. Interoperability between related systems is typically defined by a common specification. However, because of the limitations of specifications and the lack of incentives for programs to probe beyond them, the resulting interoperability is often less than desired.

To remedy this situation, management approaches and techniques that bridge the gaps between the isolated programs and perspectives are needed. Some candidate strategies for achieving programmatic interoperability include

- synchronization of schedules and budgets
- joint risk management
- coupled award fee boards
- linked promotions

Achieving programmatic interoperability in a system-of-systems context demands consistent collaboration among the various program offices. Communication must be consistent and risk must be shared and distributed across programs. Failure to build management structures that encourage programs to interoperate will only perpetuate stovepipe management practices.

4.2.2 Constructive Interoperability

Constructive interoperability addresses technologies (and the technical activities to select and apply them) that create and maintain interoperable systems. These technologies commonly include shared architectural elements, data specifications, communications protocols, and common standards. Constructive interoperability is the nuts and bolts of what we commonly think of as system and software engineering. We expect good engineers to select the right technologies and approaches and apply them in thoughtful ways.

Selecting and using some common set of technologies is necessary, but is normally not sufficient to produce highly coupled interoperation. Currently available technologies capture only part of rich semantic context and underlying assumptions regarding data that are required for sophisticated interoperation. For example, our research examined two systems intending to interoperate that used object request brokers provided by different COTS vendors. The two program offices assumed that conformance to an industry standard would ensure interoperability. Unfortunately, the vendor of one broker added unique features to the product that extended the standard. These unique features were used during construction of one of the systems, making it impossible for the two systems to interoperate without considerable rework to one or both systems.

Achieving constructive interoperability demands new perspectives on the use of standards and software architectures. It is naïve to assume that simply using a standard will guarantee system interoperability. Joint definition of the standards and system-of-systems architecture will provide a critical aspect for each system's construction.

4.2.3 Operational Interoperability

Operational interoperability refers to the activities related to the operation of a system in the context of other systems. These activities include defining

- doctrine governing the way the system is used
- conventions for how the user interprets information derived from interoperating systems (i.e., the semantics of interoperation)
- mechanisms for interacting with program offices to improve interoperability between programs

Problems in achieving interoperability at other levels inevitably lead to problems at the operational level. For example, in one case, multiple combat platforms were intended to exchange data. However, the platforms supported different communication links. Each type of link places different restrictions on the data. As a result, different users were receiving different views of the battle environment. This problem at the constructive level of interoperability also led to operational conventions for *how* the user interpreted data.

4.2.4 The Interoperability Environment

The model we posit is still incomplete, in that it does not address another set of actors who are critical to the interoperability problem and solution. These actors—particularly important within the DoD enterprise but with analogs in commercial enterprises as well—are involved in establishing high-level vision, setting policy direction, and creating or selecting enterprise-wide standards. An expansion of the model includes these actors and their activities as part of the environment in which interoperability must be achieved and maintained (Figure 6).

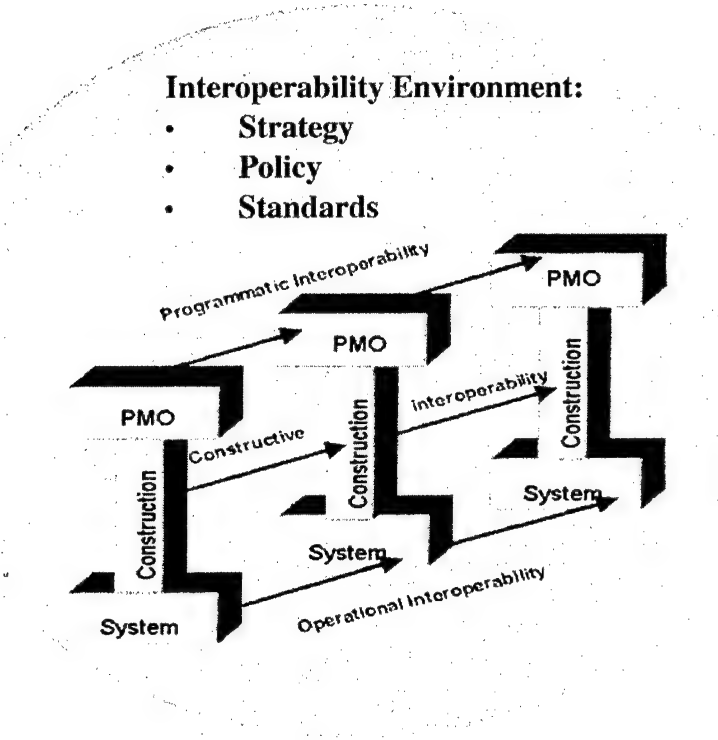


Figure 6: The Interoperability Environment

It is a mistake to think of the environment in which interoperability must be achieved as a single entity. In reality, this environment is made of many organizations that contribute high-level vision, policy, and standards. Unfortunately, according to the SOSI experts, these contributions can conflict and at times hinder efforts to achieve interoperability. In effect, there may be a lack of interoperability between various visions, policies, and standards.

4.2.5 Contributions of the SOSI model

The SOSI model was originally constructed as a mechanism for organizing our inquiry into interoperability problems. It was not intended to categorize interoperation among systems in the sense of the LISI or NATO models, and therefore should not be viewed as a competitor to those models. It does not provide the detail of the other models, and must be significantly

enhanced to model or account for the many aspects and elements that bring about or prevent interoperability in any given instance.

The contribution of the SOSI model to the discourse is seen in the emphasis on *activities that must be performed by various actors* in order to achieve the interoperability goal. In a sense, it is an attempt to create a bridge between the interoperability problem (achieving interoperability at some expected level) and solution (presumably involving improved practices and technologies). The SOSI model identifies four major groups that will have to be part of any eventual solution:

- leadership that is responsible for creating strategy, setting policy, and establishing standards
- program offices responsible for management of interoperating systems
- organizations that select architectures and technologies and implement the systems
- operational entities that participate in the definition of requirements, and eventual use, of the system of systems

4.3 Characteristics of Interoperability

We have already suggested that interoperability is a multi-dimensional phenomenon. Ultimately, useful processes and models must account for many more characteristics of interoperability than are addressed by the union of today's models, including the SOSI model. The SEI has begun work on a first step in this process by defining and categorizing a set of characteristics of interoperability that apply to components, systems of systems, and integration mechanisms.

The characteristics identified to date do not form a complete set and the relationships among elements of the set are poorly understood. However, we include the following snapshot of the categorization in order to stimulate interest in the many aspects of interoperability that must be analyzed and modeled.

Table 1: Characteristics of Interoperability

Category	Description
Communication	Characteristics that address the transmission and reception of messages (including exchanged or shared data) between nodes.
Data Management	Characteristics that address the storage, distribution, and synchronization of data, including what data are exchanged and how data flow between nodes.
Dynamism	Characteristics that address how capabilities or services are made available, identified and located, and composed into composite systems on the fly or at runtime.
Evolution	Characteristics that address how individual nodes, the integration mechanisms binding nodes together, and the resulting composite systems will change over their respective lifetimes. Characteristics include expected changes and rates of changes, relationships between rates of changes, and the expected lifetime of the whole and its parts.
Semantic Consistency	Characteristics that capture the degree of agreement on meaning of communicated information, functionality, and transformations. A slightly more general point of view is that characteristics in this set will describe to which degree nodes share the same world view.
Ownership	Characteristics relating to who the controlling authorities are for nodes, systems, and processes. In many cases there is single entity that completely controls all nodes in a system. In other circumstances there may be no single entity that is responsible for the overall composite system.
Usage Patterns	Characteristics related to the expected pattern of use for nodes and composite systems. Usage patterns of the composite, interoperating system will be harmonious or will conflict with the collection of usage patterns of its constituent parts. Generally, greater complexity and heterogeneity in nodes implies less consistency and harmony in usage patterns.

Our intent is to continue to build and refine the set of characteristics and their classification. We believe this work is essential in order to

- fully understand how to classify different types of components, integration mechanisms, and systems of systems
- develop diagnostic instruments to identify potential problems in systems of systems

- develop predictive instruments that allow us to understand how well a component will perform within a system of systems and how the system itself will perform

As we begin work to address these tasks, we will keep clear sight of our basic principles[4] and expand the reach of our model.

[4] These principles include: little distinction between system and systems of systems, domain independence, imperfection information, no one-time solution, centrality of legacy, and emergent properties.

5 Current and Ongoing Work at the SEI

The SEI is not alone in recognizing the interoperability problem. It has been apparent to many organizations, including developers, acquirers, and most other types of software-dependent enterprise. Anyone who builds, maintains, or uses complex systems has encountered the problem for some time. The entire software community is well aware that the problems of interoperability are many and are not likely to disappear quickly.

And, in truth, many effective strategies have been devised that have produced successfully interoperable systems. For example, the software engineering community has developed processes to reach consensus on interoperability requirements, even in the awareness that they must be fluid; we have also devised methods to carefully manage their change. We can put in place a common oversight organization as the arbiter of conflicts of interest, as has occurred with the military's Joint Forces Command. We can impose standards even in the knowledge that some components will suffer as a result.

But what is changing today are the greatly increased expectations for interoperability in every domain. If the computerization of specific technical and business operations represents an initial great leap in enhancing access to enterprise data, then the fusing of that data across systems into seamless streams of information represents another—and a much greater—step. Consider the following speculative scenario that appeared in *Scientific American*:

> At the doctor's office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom's *prescribed treatment* from the doctor's agent, looked up several lists of *providers*, and checked for the ones *in-plan* for Mom's insurance within a *20-mile radius* of her *home* and with a *rating* of *excellent* or *very good* on trusted rating services. It then began trying to find a match between available *appointment times* (supplied by the agents of individual providers through their Web sites) and Pete's and Lucy's busy schedules.[5]

Comparable scenarios, where various computer systems interoperate in a complex and dynamic manner, are being created in almost every domain, from e-business to industrial control to coalition combat.

---

5 The Semantic Web is the name attached to a future generation Web capability where information is given well-defined meaning, better enabling computers and people to work in cooperation [Berners-Lee 01].

As might be expected, many industrial, commercial, and military organizations are investigating the approaches that will bring these scenarios to reality. The research study by the SEI identified dozens of efforts to improve the interoperability of joint and coalition military systems. A Web search will convince the reader that there are many more community efforts just to establish interoperability standards, and an unknown (and very likely a large) number of efforts within individual commercial organizations.

The primary mission of the SEI is to act as an agent for technology transition. The SEI's ISIS initiative will therefore seek to provide a focal point for sharing information and new technologies among many scattered communities of interest. During the coming year, we expect that work will be done, both by us and others in the software engineering community, to

- codify current management and engineering practices for construction and evolution of systems of systems
- identify and characterize techniques for forming and evolving systems of systems
- develop evaluation approaches for selecting constituent elements of systems of systems
- analyze current and emerging technologies and products with potential applicability to the integration and interoperability of systems of systems
- develop cost models

For this work, our collaborators include Aerospace, Inc., the SEI's Software Engineering Measurement and Analysis (SEMA) group, the University of North Carolina, and Defense Acquisition University. Our sponsors include the Deputy Assistant Secretary of the Navy for Acquisition Management DASN(Acq) and the Office of the Secretary of Defense Program Analysis and Evaluation Cost Analysis Improvement Group (OSD/PA&E CAIG).

Taking a longer perspective, over the next several years, ISIS will

- establish an independent perspective looking across the diverse communities working on interoperability problems
- assist in standardization efforts, particularly where we can play an effective role representing the perspective of our primary clients
- provide independent and reasoned analysis regarding the effectiveness of policies and programs, and the readiness of technologies
- act as a transition agent to move new ideas from individual communities into wider practice

In the immediate future, we will focus on gathering information about large system-of-system efforts being built or maintained by government clients. We believe ISIS has a strong foundation to address the interoperability problems of the community. From the initial research project in interoperability, we have a good understanding of the problems and the important in-

sight of addressing those problems from programmatic, constructive, and operational perspectives. From our COTS-Based Systems heritage, we bring an appreciation of the complexities of making diverse systems constructed by independent organizations work together. We also inherit recognition that any successful way forward will embrace the fact that, no matter how successful the community is in creating new standards, the remaining uncertainty will demand flexible strategies for system-of-systems construction. Working with colleagues at the SEI will help us better understand the requirements and technologies needed for performance-critical systems and architectures, as well as the processes that can build and maintain powerful and flexible systems of systems.

We have already begun efforts to engage the government community in our tasks' achievement. We look forward to similar activities with the commercial, industrial, and academic communities.

Bibliography

[Audiworld 02] Audiworld. "The New Audi A8: A New Sporting Dimension in the Luxury Segment." *Audiworld*, Sept 8, 2002. <http://www.audiworld.com/news/02/a8launch/content6.shtml>.

[Berners-Lee 01] Berners-Lee, Tim; Hendler, James; & Lassila, Ora. "The Semantic Web." *Scientific American 284*, 5 (May 2001). <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>.

[Brooks 87] Brooks, Fred. "No Silver Bullet: Essence and Accidents of Software Engineering." *IEEE Computer 20*, 4 (April 1987): 10-19.

[Brunnermeier 99] Brunnermeier, Smita B. & Martin, Sheila A. *Interoperability Cost Analysis of the U.S. Automotive Supply Chain*. National Institute of Standards & Technology, March 1999. <http://www.nist.gov/director/prog-ofc/report99-1.pdf>.

[C4ISR 98] C4ISR Architecture Working Group. *Levels of Information Systems Interoperability (LISI)*. U.S. Department of Defense, March 1998. <http://www.defenselink.mil/nii/org/cio/i3/lisirpt.pdf>.

[DoD 01] Department of Defense. Department of Defense Dictionary of Military and Associated Terms. Washington, DC: Operational Plans and Joint Force Development Directorate, 2001. <http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf>.

[Goldwater 86] Goldwater-Nichols Department of Defense Reorganization Act of 1986 (Public Law 99-433). <http://www.dtic.mil/jcs/core/title_10.html>.

[Levine 03] Levine, L.; Meyers, C.; Morris, E.; Place, P.; & Plakosh, D. *Proceedings of the System of Systems Interoperability Workshop (February 2003)* (CMU/SEI-2003-TN-016). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. <http://www.sei.cmu.edu/publications/documents/03.reports/03tn016.html>.

[Meyers 01a] Meyers, B. Craig & Oberndorf, Patricia. *Managing Software Acquisition—Open Systems and COTS Products* Boston, MA: Addison-Wesley, 2001.

[Meyers 01b] Meyers, B. Craig & Oberndorf, Patricia. *A Framework for the Specification of Acquisition Models* (CMU/SEI-2001-TR-004, ADA401717). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports/01tr004.html>.

[NATO 00] ISSC NATO Open Systems Working Group. "NATO C3 Technical Architecture Volume I – C4ISR/Sim TRM Applicability to NATO Interoperability – Management." Version 4.0, March 2000. <http://nc3ta.nc3a.nato.int/volume1/>.

[NCS 96] National Communications System. *Telecommunications: Glossary of Telecommunication Terms* (Federal Standard 1037C). Arlington, VA: National Communications System, 1996. <http://www.its.bldrdoc.gov/fs-1037/>.

[Standards 90] Standards Coordinating Committee. *IEEE Standard Glossary of Software Engineering Terminology* (IEEE Std 610.12-1990). New York, NY: The Institute of Electrical and Electronics Engineers, 1990.

[Tolk 03] Tolk, Andreas & Muguira, James A. "The Levels of Conceptual Interoperability Model." *Proceedings of the 2003 Fall Simulation Interoperability Workshop*. Orlando, Florida, Sept. 14-19, 2003. Orlando, FL: Simulation Interoperability Standards Organization, 2003.

[Warner 02] Warner,Neil. "Interoperability - An Australian View." *Proceedings of the 7th International Command and Control Research and Technology Symposium.* Quebec City, Canada, Sept. 16-20, 2002. Washington, DC: CCRP/Department of Defense, 2002.

[Wilson 03] Wilson, Robert. "Audi A8." *The Weekend Australian*, October 30, 2003. <http://www.theaustralian.news.com.au/common/story_page /0,5744,7706846%255E13232,00.html>.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE March 2004	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Current Perspectives on Interoperability		5. FUNDING NUMBERS F19628-00-C-0003
6. AUTHOR(S) Lisa L. Brownsword, David J. Carney, David Fisher, Grace Lewis, Craig Meyers, Edwin J. Morris, Patrick R. H. Place, James Smith, Lutz Wrage		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2004-TR-009
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESC-TR-2004-009
11. SUPPLEMENTARY NOTES		
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

This report describes current research within the software engineering community on the topic of interoperability between software systems. That research includes analyses of the different types of interoperability problems and issues and efforts to define models of interoperability that will aid in creating solutions to those problems.

The report also describes work that is currently underway at the Software Engineering Institute (SEI) in this area. That work originated in an independent research effort and now has grown into a separate technical initiative in the area of interoperability. The SEI initiative is currently focused on analyzing several aspects of interoperability: how it is manifest in different kinds of activities (i.e., programmatic vs. constructive vs. operational activities), the essential characteristics of interoperability, and the key principles on which solutions will depend.

14. SUBJECT TERMS integration, interoperability, systems of systems, models			15. NUMBER OF PAGES 50
16. PRICE CODE			
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL